V-ZED — THREE NEW FUNCTIONS

This is a regular feature to assist VZ 200 users to come to understand more about their computers and to learn a few tricks which are not necessarily covered by the manuals. We welcome contributions from Readers who have discovered new features of the machine or interesting techniques which they would like to share with their fellow VZ-200 users.

The BASIC interpreter in the VZ 200 was written by MICROSOFT, the company which developed the first BASIC interpreter for a microcomputer way back in the mid 70's and which probably supplies over 80% of all BASIC interpreters in use today. Not surprisingly, when a new computer such as the VZ comes along, MICROSOFT takes its standard BASIC interpreter and modifies it to suit the new hardware and the particular features which the manufacturer would like included. From the user's point of view there are both advantages and disadvantages to this approach. The main disadvantage is that the resulting code can become very untidy with patches on patches right throughout the ROM. The outcome often being inefficient use of space and slower execution. On the positive side however, there are likely to be routines still left in from other interpreters which are not intended to be available in the VZ but, with a little fiddling can be used. To the average computer user, the thrill of making your computer do something which the manufacturer never intended, is worth any of the disadvantages. The purpose of this article is to start you off with three hidden functions. Once you start experimenting in this area you will no doubt find others. Please write in and let us know about them so that we may all share in them.

The MICROSOFT BASIC interpreter as implemented in the Tandy TRS-80 Model 1 occupied 12 Kbytes of ROM. Although we do not know for sure, it is likely that this implementation started a new family of BASIC interpreters of which the VZ's is a derivative. Certainly there seems to be no surplus code in the Tandy interpreter although the Model 3 version shows evidence of having been extensively patched and hacked around. The interpreter in the VZ has a number of additional features over and above those available in the Tandy. In particular, the support for higher screen resolution, colour and full screen editing obviously requires extra code. Even though this interpreter now occupies 16K of ROM it became necessary to leave out some of the features which had been in the TRS-80 version. In particular, the AUTO TRACE function and the free memory indicator have gone whilst there is no facility to turn off the sound, should you wish to do so. However, the essential routines to do all these things remain locked away in the ROM and can be accessed with a bit of judicious POKEing.

AUTO LINE NUMBERING

The interpreter contains an AUTO line numbering routine which when activated, automatically prints the next line number on the screen to speed up the entry of BASIC programs. It is possible to specify the starting line number and the increment between line numbers. For example, you may wish to start entering lines commencing with line 100 with an increment of 10 so that the second line would be 110 the third 120 etc. The AUTO routine operates every time you press the RETURN key from the COMMAND mode. It looks at address 30945. If that address contains a zero then AUTO numbering is off and the computer behaves normally. However, if that value is 1, the AUTO routine looks at addresses 30946 and 30947 to find the value of the starting line number then at addresses 30948 and 30949 for the increment between line numbers. The next line number is then automatically displayed on the screen. The only part of the AUTO routines missing is the ability to recognise the AUTO command itself. However, if you POKE the appropriate values into the memory addresses above, you will be able to use this facility.

To set the starting line number, POKE the decimal equivalent of its Least Significant Byte (LSB) into address 30946 and the decimal equivalent of its Most Significant Byte (MSB) into 30947. Similarly, to set the line increment, POKE its LSB into 30948 and its MSB into 30949. It is likely that this is double Dutch to relatively new users of the VZ so we have illustrated the techniques with the program below. If you wish to know more about the subject of POKEing etc. you will find a good article in Volume 4, Issue 4/5.

We suggest you enter this routine, make sure it works satisfactorily then CSAVE it under the name AUTO or similar. You can then load it in whenever you are doing program development. We have used high line numbers to keep it out of the way of your own programs. To start it operating, type RUN 60,000. Incidentally, you terminate AUTO line numbering by pressing the BREAK key.

PROGRAM LISTING 1

```
60000 REM SET STARTING LINE NO                    FOR THE AUTO ROUTINE
60010 INPUT"STARTING LINE NUMBER";SL
60020 POKE 30946,(SL-256*INT(SL/256))
60030 POKE 30947,INT(SL/256)
60050 REM SET THE INCREMENT                       BETWEEN LINE NUMBERS
60060 INPUT"INCREMENT BETWEEN LINE NOS";IN
60070 POKE 30948,(IN-256*INT(IN/256))
60080 POKE 30949,INT(IN/256)
60100 REM SWITCH ON THE AUTO                      LINE NUMBERING ROUTINE
60110 POKE30945,1
```

TURNING OFF THE BEEPING KEYBOARD

Now that you have AUTO line numbering, you will probably want to sit up all night entering programs. Only trouble is, the beeping of the keys is likely to keep the rest of the family awake.

No problem:

POKE 30779, 0 disables the key beep whilst

POKE 30779, 1 turns it on again.

You may enter this straight from the keyboard or include it as a line in your program.

Incidentally, this memory address appears to carry out some other functions, depending on the bit that is set. We did a little experimenting and found that bit 0 turns on and off the beep as expected i.e. an even value POKEd into address 30779 turns off the beep whilst an odd number turns it on i.e. 0, 2, 4, 6, 8 etc. turn it off, 1, 3, 5, 7, 9 etc. turn it on. Bits 1 and 2 have no special effect but bit 3 clears the screen and positions the cursor at the bottom left hand corner. This bit also causes an audible click from somewhere inside the computer probably from the piezo electric speaker. Bit 4 changes the background colour from green to orange. As far as we could tell bits 5, 6 and 7 had no effect.

FREE SPACE

Probably the most useful POKE for a programmer would be a way of finding out how much string space is available or how much memory you have left to cram in those last few lines before being told by the machine that you are Out of Memory.

Try the following.

```
POKE 30862,212:
POKE 30863,39:
PRINT USR(X) 'FREE MEMORY
  OR
PRINT USR(X$) 'FREE STRING
SPACE
```

Micro-80 Jul 84 4(7) : p 3-4.